

Evaluation formelle de la Qualité de Service pour les systèmes d'acquisition

Belgacem Ben Hedia, Fabrice Jumel, Jean-Philippe Babau

CITI - INSA Lyon – CPE, F69621 Villeurbanne Cedex - FRANCE

e_mail : {belgacem.ben-hedia ; fabrice.jumel; jean-philippe.babau}@insa-lyon.fr

Résumé

Dans le domaine des applications temps réel de contrôle des procédés, la validation se fonde sur une connaissance précise des caractéristiques temporelles des données utilisées comme le retard et le taux de pertes. Ces données sont fournies par un logiciel dédié appelé pilote. En conséquence, il est nécessaire d'évaluer l'impact du pilote sur la QoS (Qualité de Service) des données. Ce travail propose un modèle formel des pilotes d'acquisition de données basé sur les automates temporisés communicants et montre l'influence des paramètres du pilote sur la QoS fournie par le système d'acquisition.

MOTS-CLÉS : temps réel, automates temporisés, pilotes d'équipements, validation, qualité de service

1. Introduction

L'utilisation de l'informatique dans le cadre du contrôle des procédés est de plus en plus courante. Elle permet d'implémenter des lois de contrôle utilisant les données délivrées par les capteurs pour produire des commandes au niveau des actionneurs.

Actuellement ces systèmes sont de plus en plus complexes et la réutilisation des composants logiciels doit permettre de réduire les coûts de développement. Dans le domaine du contrôle des procédés, les composants réutilisables sont classiquement des services fournis par un système d'exploitation temps réel respectant les normes OSEK [ZAH 98] ou POSIX [ISO 96]; des services de communication distants (FT layer de TTA [TTA 99]); et enfin des services de communication avec le procédé contrôlé via des pilotes d'équipements (systèmes d'entrée/sortie). Le but de ce pilote d'équipement et de fournir un interface entre le capteur physique et l'application. Dans ce travail nous considérons les pilotes d'acquisition des données des systèmes de contrôle des procédés [FOK 02]

Due aux contraintes de QoS inhérentes a ces systèmes, il est nécessaire de pouvoir caractériser la QoS du pilote. Dans ce travail nous considérons trois critères classiques de QoS pour un systèmes de

contrôle des procédés : retard minimum et maximum, le pourcentage de perte maximal, le nombre maximal de pertes consécutifs.

Les techniques le plus connues pour calculer un retard maximum sont basés sur un analyse d'ordonnancement des tâches et des messages [MIG 03]. Ces techniques considèrent des architectures simples, généralement statiques, et produisent des bornes généralement non réalistes. Dans le but de remédier a ce problème, plusieurs travaux se basent sur des techniques permettant d'avoir une modélisation exhaustive du comportement temporel des systèmes. Il s'agit en particulier d'utiliser les automates temporisés communicants ou les automates hybrides [BEL 04], [DAV 03] [VES 00].

Nous avons choisi d'utiliser ce type d'approche pour l'étude de la QoS des pilotes (à l'aide de l'outil IF [BOZ 99], [BOZ 02]). Ceci permet d'avoir des caractéristiques temporelles plus réalistes de l'application pour des architectures plus complexes (prise en compte du comportement dynamique).

Le reste de l'article est organisé de la manière suivante. La partie suivante présente le système d'acquisition des données et les critères de QoS à évaluer. La troisième partie présente la stratégie de modélisation. Enfin la dernière partie illustre l'approche en analysant l'impact de paramètres de systèmes d'acquisition sur les critères de QoS. Nous terminons par une conclusion générale et quelques perspectives.

2. Contexte

2.1. Système d'acquisition des données

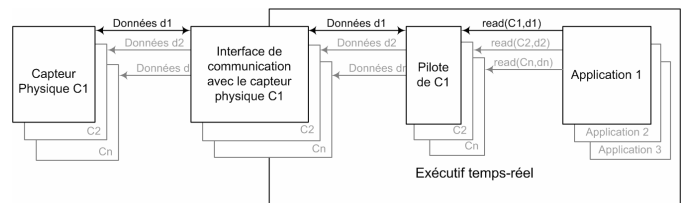


Figure 1. Système d'acquisition de données

Un système d'acquisition des données (c.f figure 1) est composé de quatre éléments:

- **Capteur physique:** Il permet de convertir les informations (température, vitesse) produites par l'environnement en données numériques.

- **Interface de communication:** Il connecte le capteur à la partie logiciel de système. Il récupère les données produites par le capteur physique et la sauvegarde dans des registres accessibles par le pilote.

- **Pilote d'équipement:** C'est un logiciel dédié intégré au système d'exploitation et indépendant de l'application. Il permet l'abstraction de la couche matérielle. Il permet à l'application d'accéder aux données.

- **L'application temps réel:** une application temps réel récupère les données de pilote.

On distingue ici le pilote matérielle (appelé ici interface de communication) et le pilote logiciel (appelé ici pilote d'équipement).

2.2. Définition des critères de QoS

Suivant les applications, les critères de QoS les plus appropriés peuvent être différents [RAM 99]. Pour la suite, les critères de QoS considérés sont:

- **Retard:** les données utilisées par l'application reflètent bien l'état du procédé mais à un instant plus ou moins passé. Nous considérons le **retard minimum** et le **retard maximum**.

- **Taux de perte:** certaines informations ne seront jamais rendues disponibles pour l'application et sont donc perdues. Le taux de perte est le pourcentage des données perdu par rapport aux données produites. Nous considérons uniquement le **taux de perte maximum**.

3. Modèles formels de système d'acquisition de données.

3.1. Principe de Modélisation du système

Cette partie présente le principe de modélisation de système d'acquisition de données en IF:

- Chaque élément de système est modélisé par un processus IF (c.f. figure2).
- Le modèle de communication étant asynchrone en IF, une opération lecture synchrone est modéliser par une émission de signal *getxx()* ou *read()* et l'attente d'un signal *retxx()* (c.f figure 2).
- Seul le comportement temporel est modélisé, les valeurs des données sont utilisées à des fins d'observation sur le modèle.

- Nous considérons trois lois de production de données par le capteurs: périodique (avec ou sans gigue), sporadique, ou en rafale [MEM 98].
- Le registre du stockage de données est de type FIFO ou LIFO.

Puisque la durée d'une transition en IF est nulle, l'évolution du temps est considérée uniquement à travers les gardes temporelles sur les horloges. Ces gardes peuvent représenter un intervalle discret par l'intermédiaire de mot clé IF (*delayable*). Due à la sémantique temporelle de IF, modéliser une durée d'exécution (relier à la durée d'exécution et d'attente de processeur d'une tâche) peut être obtenue grâce à l'ajout d'un état d'attente dans les processus concerné.

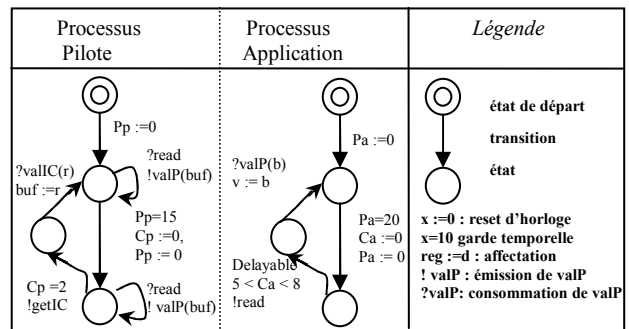


FIG. 2 – Modèle de système d'acquisition pour une émission périodique (période P_c de 10), un pilote à scrutation (période P_p de 15, durée C_p de 2) et une application en mode scrutation (période P_a de 20, durée C_a de 6 à 7)

3.2. Modélisation de la QoS

Pour évaluer les critères de QoS nous proposons deux approches.

La première consiste à introduire des variables d'observation dans le modèle. Les variables sont des compteurs d'instance des signaux (reçu par l'application, produite par le capteur) et des compteurs temporels (date de production et de consommation des données).

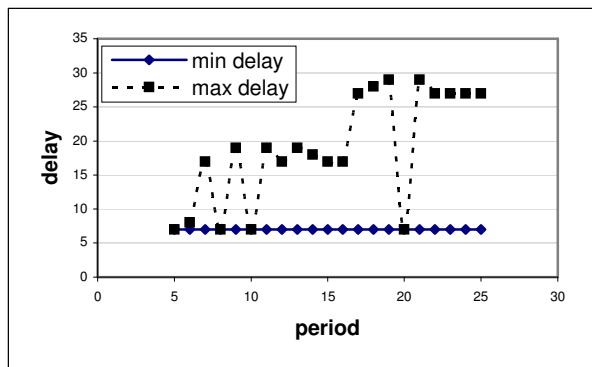
La deuxième approche consiste à introduire des processus observateurs dans le modèle. Un observateur est un processus spécifique qui modélise un critère de QoS.

4. Expérimentations

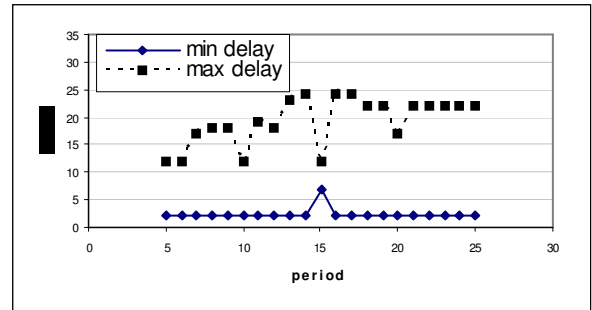
Dans le but d'évaluer l'intérêt de l'approche proposée, on considère les hypothèses suivantes et les résultats sont présentés dans les figures 3.x:

- La période de capteur physique P_c à 10 unités de temps.
- Le registre de l'interface de communication est de taille 1.
- Le pilote est en mode scrutation et sa période P_p est inconnu a priori, la taille de son registre est de 1 avec écrasement de donnée si nécessaire, il a un temps de réponse égale a 2 unité de temps pour (3a, 3c, 3b) et égale à [1,2] pour (3d).
- La période de l'application P_a est égale à 20 unités de temps pour (3a, 3c, 3d) et égale à 15 unités de temps pour (3b) et son temps de réponse est égale à 7 unité de temps pour (3a, 3b) et égale à [6,7] pour (3c, 3d).
- En fin, nous considérons que tous les processus du système (capteur, pilote et application) commencent au même instant.

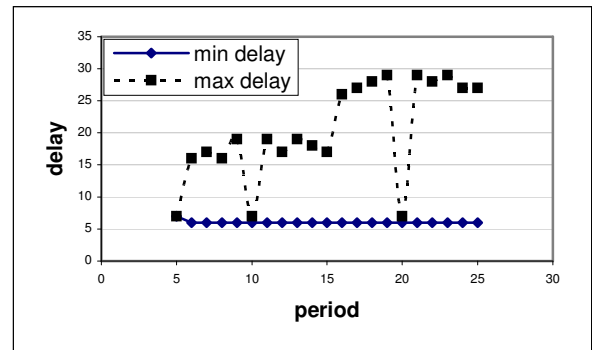
Dans la figure 3a. Le retard minimal est de 7 (lecture toutes les $(20k + 7)$ unités de temps d'une donnée à priori produites par le capteur toutes les $(20k)$ unités de temps). Quelques points remarquables demeurent, correspondant à un séquençement particulier, par exemple pour 10 et 20 où la synchronisation entre l'application, le capteur et le pilote amène à l'existence d'un unique retard possible de 7 unités de temps. Les courbes de la figure 3c sont plus lissées, on retrouve comme uniques points particuliers, les points de synchronisation 5, 10 et 20. La figure 3d, est très semblable à la figure 3c, à part quelques petites différences de ± 1 liés à l'influence de l'intervalle [1,2]. Enfin sur la figure 3b, le fait de changer la période de l'application de 20 unités de temps à 15 unités de temps entraîne des changements considérables au niveau, bien sûr, des valeurs obtenues, mais aussi de la forme de la courbe.



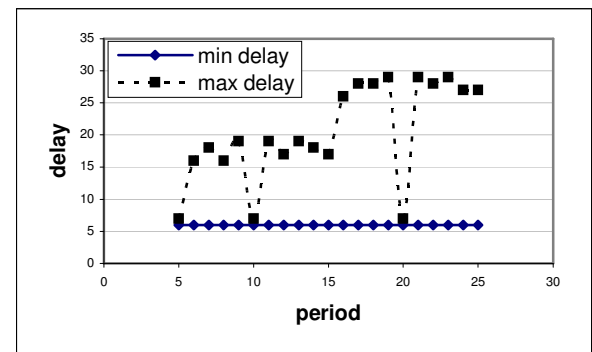
(a) Temps de réponse du pilote : 2
Période application : 20, Temps de réponse : 7



(b) Temps de réponse du pilote 2
Période application 15, Temps de réponse : 7



(c) Temps de réponse du pilote 2
Période application 20, Temps de réponse dans [6,7]



(d) Temps de réponse du pilote [1 2]
Période application 20, Temps de réponse dans [6,7]

Figure 3. Evolution du retard vis-à-vis de la période de scrutation du pilote pour différentes configurations du système

4.1. Analyse

Nous avons sur cet exemple montré la grande richesse d'expression des modèles proposés. En particulier, il est possible de prendre en compte l'existence de propriétés temporelles définies sous forme d'intervalles. Les résultats des analyses analytiques n'ont généralement pas de formes closes et ne présentent donc pas d'intérêt particulier vis-à-vis de ceux obtenus par analyse exhaustive.

L'intérêt de l'analyse exhaustive est de concentrer le travail sur la création des modèles alors que pour une analyse classique un travail important doit être fait pour l'obtention des résultats. Bien entendu, le problème majeur de l'analyse exhaustif est le risque d'explosion combinatoire. Le tableau 1 présente l'évolution du nombre d'états du graphe généré (sans compteurs). Les exemples a et b traités au niveau de la figure 5 ne posent pas de difficultés. On remarque, par contre, qu'un exemple plus complexe incluant une non périodicité de la production augmente considérablement le nombre d'états du graphe généré. Cet exemple n'a donc pas pu être traité car l'analyse des modèles (en rajoutant les compteurs) explose sur la machine de faible puissance utilisée pour réaliser les expérimentations

Exemple	<i>a</i>	<i>b</i>	<i>C</i>
période de production	10	10	[10,11]
période du pilote	7	7	7
durée pilote	2	[1,2]	[1,2]
Période de l'application	20	20	20
durée de l'application	7	[6,7]	[6,7]
nombre d'états (sans compteur)	387	1676	19331

TAB.1 – Evolution du nombre d'états en fonction de la complexité de l'exemple

5. Conclusion

Nous avons proposé des modèles de spécification comportementale et temporelle d'un système d'acquisition de données. Ces modèles permettent de caractériser un pilote d'acquisition du point de vue de la qualité de service. Ces modèles ont été réalisés en IF et prennent en compte le capteur, l'interface de communication, le pilote et l'application. Les résultats présentés ont montré que la forme des propriétés de la qualité de service est complexe et dépend de plusieurs paramètres.

Les perspectives de cette étude portent sur la prise en compte d'architectures plus complexes (multitâche, multiplexage des données de plusieurs capteurs, structuration en couches) et sur l'utilisation de formalismes plus riches (temps continu, modèles stochastiques) permettant d'évaluer d'autres critères de QoS.

Les problèmes d'explosion combinatoire soulevés aux paragraphes précédents feront bien entendu l'objet d'une étude approfondie. Nous évaluerons les réductions par équivalence et/ou l'introduction de résultats analytiques évalués a priori.

Références

- [BOZ 99] M. Bozga Symbolic verification for communication les protocols. Phd thesis, Verimag, University Joseph Fourier, 1999.
- [BEL 04] M. Belarbi, J.-P. Babau, J.-J. Schwarz, "Temporal Verification of Real-Time Multitasking Application Properties Based on Communicating Timed Automata", Proc. in 8-th IEEE International Symposium on Distributed Simulation and Real Time Applications, Budapest, 2004.
- [ZAH 98] A. Zahir, P. Palmieri, "OSEK/VDX-operating systems for automotive applications", in OSEK/VDX Open Systems in Automotive Networks, 1998
- [BOZ 02] M. Bozga, S. Graf, L. Mounier, "IF-2.0: A Validation Environment for Component-Based Real-Time Systems", In Ed Brinksma, K.G. Larsen (Eds.) Proceedings of CAV'02 (Copenhagen, Denmark) LNCS vol. 2404 Springer-Verlag July 2002.
- [DAV 03] A. David and all "A tool architecture for the next generation of uppaal", Technical report, Uppsala University, 2003.
- [FOK 02] F.J. Fokkink, and all "Refinement and verification applied to an in-flight data acquisition unit", in Proc. 13th Conference on Concurrency Theory - CONCUR'02, Brno, Lecture Notes in Computer Science 2421, pp. 1-23, Springer, 2002.
- [ISO 96] ISO/IEC Standard 9945-1: 1996 [IEEE/ANSI Std 1003.1, 1996 Edition] Information Technology—Portable Operating System Interface (POSIX)—Part 1: System Application: Program Interface (API) [C Language]. IEEE ISBN: 1-55937-573-6, 1996.
- [TTA 99] Time-Triggered Technology TTTech Computertechnik AG. "Specification of the TTP/C protocol". ,Technical report, Vienna, Austria, July 1999.
- [VES 00] S. Vestal "Modeling and Verification of Real-Time Software Using Extended Linear Hybrid Automata" Proc Fifth NASA Langley Formal Methods Workshop 2000.
- [MEM 98] Z. Mammeri., "Expression and derivation of temporal constraints on real time applications". In: "Journal Européen des Systèmes Automatisés", JESA-APII, Hermès, V. 32 N. 5-6, p. 609-644, 1998. (In French)
- [MIG 03] J. Migge, A. Jean-Marie, N. Navet "Timing analysis of compound scheduling policies : Application to posix1003.b" Journal of Scheduling, Kluwer Academic Publishers, 6 (5), 457-482, 2003.
- [RAM 99] P. Ramanathan,,. "Overload Management in Real-Time Control Applications Using (m, k)-Firm Guarantee". IEEE Transactions on Parallel and Distributed Systems 10(6): 549-559, 1999